

日本国特許庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日  
Date of Application: 2003年12月19日

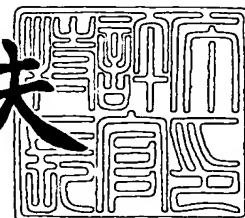
出願番号  
Application Number: 特願2003-423583  
[ST. 10/C]: [JP2003-423583]

出願人  
Applicant(s): 株式会社デンソー

2004年 1月23日

特許庁長官  
Commissioner,  
Japan Patent Office

今井康夫



出証番号 出証特2004-3002223

【書類名】 特許願  
【整理番号】 PY20032219  
【提出日】 平成15年12月19日  
【あて先】 特許庁長官殿  
【国際特許分類】 G06F 11/28  
【発明者】  
    【住所又は居所】 愛知県刈谷市昭和町 1 丁目 1 番地 株式会社デンソー内  
    【氏名】 東道 徹也  
【発明者】  
    【住所又は居所】 愛知県刈谷市昭和町 1 丁目 1 番地 株式会社デンソー内  
    【氏名】 岩井 明史  
【特許出願人】  
    【識別番号】 000004260  
    【氏名又は名称】 株式会社デンソー  
【代理人】  
    【識別番号】 100068755  
    【弁理士】  
    【氏名又は名称】 恩田 博宣  
【選任した代理人】  
    【識別番号】 100105957  
    【弁理士】  
    【氏名又は名称】 恩田 誠  
【先の出願に基づく優先権主張】  
    【出願番号】 特願2003- 54082  
    【出願日】 平成15年 2月28日  
【手数料の表示】  
    【予納台帳番号】 002956  
    【納付金額】 21,000円  
【提出物件の目録】  
    【物件名】 特許請求の範囲 1  
    【物件名】 明細書 1  
    【物件名】 図面 1  
    【物件名】 要約書 1  
    【包括委任状番号】 9908214

**【書類名】 特許請求の範囲****【請求項 1】**

制御要求仕様が要求仕様記述言語にて記述された制御モデルから所定のプログラム言語にて記述された制御プログラムを自動生成する自動コード生成手段を用いて該所定のプログラム言語にて記述された制御プログラムを検査する制御プログラムの検査方法において、

前記制御モデルの挙動をシミュレーションするシミュレーション手段の動作結果と前記制御プログラムを実行するプログラム実行手段の動作結果とのそれぞれ対応する動作結果毎にこれらに関連付けして出力して、前記制御モデル及び前記制御プログラムの少なくとも一方の異常の有無を検査する

ことを特徴とする制御プログラムの検査方法。

**【請求項 2】**

前記関連付けが、前記自動コード生成手段による制御プログラムの自動生成に際して取得される前記制御モデルと該制御モデルから生成される制御プログラムとの対応関係を表す対応情報に基づいて行われる

請求項 1 記載の制御プログラムの検査方法。

**【請求項 3】**

前記関連付けが、前記制御モデル及び前記制御プログラムのいずれか一方に対して動作の停止箇所を指定するブレイクポイントが設定されることに基づき、他方に対しても前記対応情報に基づきその対応するブレイクポイントが設定されることによって行われる

請求項 2 記載の制御プログラムの検査方法。

**【請求項 4】**

前記ブレイクポイントが、前記制御モデルを構成する各機能ブロック毎に設定可能とされる

請求項 3 記載の制御プログラムの検査方法。

**【請求項 5】**

前記各動作結果についてのそれぞれ関連付けされた動作結果の出力が、前記プログラム実行手段により前記制御プログラムを 1 行ずつ実行及び停止させることによって行われる

請求項 2 記載の制御プログラムの検査方法。

**【請求項 6】**

請求項 2～5 のいずれかに記載の制御プログラムの検査方法において、

前記異常の有無の検査が、前記関連付けされた前記シミュレーション手段の動作結果と前記プログラム実行手段の動作結果とを逐次比較する比較手段によって行われる

ことを特徴とする制御プログラムの検査方法。

**【請求項 7】**

前記比較手段によるシミュレーション手段の動作結果と前記プログラム実行手段の動作結果との逐次比較が、前記対応情報に基づく前記制御モデルのシミュレーション順序と前記制御プログラムの実行順序との比較として行われる

請求項 6 記載の制御プログラムの検査方法。

**【請求項 8】**

前記比較手段によるシミュレーション手段の動作結果と前記プログラム実行手段の動作結果との逐次比較が、前記対応情報に基づき前記制御モデルのシミュレーションによって演算される変数値と前記制御プログラムの実行によって演算される変数値との比較として行われる

請求項 6 又は 7 記載の制御プログラムの検査方法。

**【請求項 9】**

前記対応情報に基づく前記制御モデルのシミュレーションによって演算される変数値と前記制御プログラムの実行によって演算される変数値との比較による異常の有無の検査が、前記制御モデルのシミュレーションによって演算される変数値と前記制御プログラムの実行によって演算される変数値との差が許容範囲内にあるか否かによって行われる

請求項 8 記載の制御プログラムの検査方法。

**【請求項 10】**

請求項 6～9 のいずれかに記載の制御プログラムの検査方法において、

前記比較手段により異常がある旨の判断がなされたとき、その時点での前記制御モデルのシミュレーション箇所と前記制御プログラムの実行箇所とを前記検査の結果として出力する

ことを特徴とする制御プログラムの検査方法。

**【請求項 11】**

請求項 1～10 のいずれかに記載の制御プログラムの検査方法において、

前記検査に伴って前記制御モデルのシミュレーション及び前記制御プログラムの実行が停止されるとき、前記制御モデルのシミュレーションによって演算される変数値の停止箇所において保持されている値及び前記制御プログラムの実行によって演算される変数値の停止箇所において保持されている値の少なくとも一方を変更可能とする

ことを特徴とする制御プログラムの検査方法。

**【請求項 12】**

請求項 1～11 のいずれかに記載の制御プログラムの検査方法において、

少なくとも前記制御モデルの挙動をシミュレーションするシミュレーション手段の動作結果と前記制御プログラムを実行するプログラム実行手段の動作結果とを表示手段に出力して可視表示させる

ことを特徴とする制御プログラムの検査方法。

**【請求項 13】**

制御要求仕様が要求仕様記述言語にて記述された制御モデルから所定のプログラム言語にて記述された制御プログラムを自動生成する自動コード生成手段を用いて該所定のプログラム言語にて記述された制御プログラムを検査する制御プログラムの検査装置において、

前記制御モデルの挙動をシミュレーションするシミュレーション手段の動作結果と前記制御プログラムを実行するプログラム実行手段の動作結果とのそれぞれ対応する動作結果毎にこれらに関連付けして出力する同期手段を備え、これら出力される動作結果に基づいて前記制御モデル及び前記制御プログラムの少なくとも一方の異常の有無を検査する

ことを特徴とする制御プログラムの検査装置。

**【請求項 14】**

前記同期手段は、前記自動コード生成手段による制御プログラムの自動生成に際して取得される前記制御モデルと該制御モデルから生成される制御プログラムとの対応関係を表す対応情報に基づいて前記関連付けを行う

請求項 13 記載の制御プログラムの検査装置。

**【請求項 15】**

前記同期手段は、前記制御モデル及び前記制御プログラムのいずれか一方に対して動作の停止箇所を指定するブレイクポイントが設定されることに基づき、他方に対しても前記対応情報に基づきその対応するブレイクポイントを設定することによって前記関連付けを行う

請求項 14 記載の制御プログラムの検査装置。

**【請求項 16】**

前記ブレイクポイントが、前記制御モデルを構成する各機能ブロック毎に設定可能とされる

請求項 15 記載の制御プログラムの検査装置。

**【請求項 17】**

前記同期手段による各動作結果の出力が、前記プログラム実行手段により前記制御プログラムを 1 行ずつ実行及び停止させることによって行われる

請求項 14 記載の制御プログラムの検査装置。

**【請求項 18】**

請求項 14～17 のいずれかに記載の制御プログラムの検査装置において、

前記同期手段により関連付けされる前記シミュレーション手段の動作結果と前記プログ

ラム実行手段の動作結果とを逐次比較することで前記異常の有無の検査を行う比較手段をさらに備える

ことを特徴とする制御プログラムの検査装置。

【請求項 19】

前記比較手段によるシミュレーション手段の動作結果と前記プログラム実行手段の動作結果との逐次比較が、前記対応情報に基づく前記制御モデルのシミュレーション順序と前記制御プログラムの実行順序との比較として行われる

請求項 18 記載の制御プログラムの検査装置。

【請求項 20】

前記比較手段によるシミュレーション手段の動作結果と前記プログラム実行手段の動作結果との逐次比較が、前記対応情報に基づき前記制御モデルのシミュレーションによって演算される変数値と前記制御プログラムの実行によって演算される変数値との比較として行われる

請求項 18 又は 19 記載の制御プログラムの検査装置。

【請求項 21】

前記対応情報に基づく前記制御モデルのシミュレーションによって演算される変数値と前記制御プログラムの実行によって演算される変数値との比較による異常の有無の検査が、前記制御モデルのシミュレーションによって演算される変数値と前記制御プログラムの実行によって演算される変数値との差が許容範囲内にないか否かによって行われる

請求項 20 記載の制御プログラムの検査装置。

【請求項 22】

請求項 18～21 のいずれかに記載の制御プログラムの検査装置において、

前記比較手段は、異常がある旨の判断がなされたとき、その時点での前記制御モデルのシミュレーション箇所と前記制御プログラムの実行箇所とを前記検査の結果として出力するものである

ことを特徴とする制御プログラムの検査装置。

【請求項 23】

請求項 13～22 のいずれかに記載の制御プログラムの検査装置において、

前記検査に伴って前記制御モデルのシミュレーション及び前記制御プログラムの実行が停止されるとき、前記制御モデルのシミュレーションによって演算される変数値の停止箇所において保持されている値及び前記制御プログラムの実行によって演算される変数値の停止箇所において保持されている値の少なくとも一方を変更可能とする

ことを特徴とする制御プログラムの検査装置。

【請求項 24】

請求項 13～23 のいずれかに記載の制御プログラムの検査装置において、

少なくとも前記制御モデルの挙動をシミュレーションするシミュレーション手段の動作結果と前記制御プログラムを実行するプログラム実行手段の動作結果とを可視表示する表示手段をさらに備える

ことを特徴とする制御プログラムの検査装置。

【請求項 25】

制御要求仕様が要求仕様記述言語にて記述された制御モデルから所定のプログラム言語にて記述された制御プログラムを自動生成する自動コード生成手段を用いて該所定のプログラム言語にて記述された制御プログラムを検査する際に用いられる検査プログラムであって、

前記制御モデル及び前記制御プログラムの少なくとも一方の異常の有無を検査すべくコンピュータを通じて実行する手段として、

a. 前記制御プログラムを自動生成する際に取得される情報から前記制御モデルと前記制御プログラムとの対応関係を表す対応情報を生成し、前記制御モデルをシミュレーションするシミュレーション手段と前記制御プログラムを実行するプログラム実行手段との双方の動作の停止箇所を前記対応情報に基づき関連付けする手段、

- b. 前記シミュレーション手段と前記プログラム実行手段との双方に対してシミュレーション及びプログラムの実行を指示する手段、
- c. 前記シミュレーション及びプログラムの実行の後、前記シミュレーション手段及び前記プログラム実行手段の双方の停止を検出する手段、
- d. 前記双方の停止を検出したとき、前記制御モデルのシミュレーションの結果と前記制御プログラムの実行の結果とを比較し、その比較結果に基づいて前記異常の有無を検査する手段、

を含むことを特徴とする制御プログラムの検査プログラム。

【請求項 26】

前記対応情報は、前記制御モデルと前記制御プログラムとの互いに対応する位置の関係を表す情報である実行位置対応情報を含み、

前記対応情報に基づき停止箇所を関連付けする手段は、前記シミュレーション手段及び前記プログラム実行手段のいずれか一方においてその動作の停止箇所を指定するブレークポイントが設定されることによって前記実行位置対応情報に基づき他方の対応する箇所を指定するブレークポイントを設定するものである

請求項 25 記載の制御プログラムの検査プログラム。

【請求項 27】

前記比較結果に基づいて異常の有無を検査する手段は、少なくとも前記制御モデルのシミュレーションの停止位置と前記制御プログラムの実行の停止位置とを前記実行位置対応情報に基づき比較することによって前記異常の有無を検査する

請求項 26 記載の制御プログラムの検査プログラム。

【請求項 28】

前記対応情報は、前記制御モデルの処理にかかる変数値と前記制御プログラムの処理にかかる変数値との対応関係を表す変数対応情報を含み、

前記比較結果に基づいて異常の有無を検査する手段は、少なくとも前記制御モデルのシミュレーションによって演算される変数値と前記制御プログラムの実行によって演算される変数値とを前記変数対応情報に基づき比較することによって前記異常の有無を検査する

請求項 25～27 のいずれかに記載の制御プログラムの検査プログラム。

【請求項 29】

制御要求仕様が要求仕様記述言語にて記述された制御モデルから所定のプログラム言語にて記述された制御プログラムを自動生成する自動コード生成手段を用いて該所定のプログラム言語にて記述された制御プログラムを検査する際に用いられる検査プログラムであって、

前記制御モデル及び前記制御プログラムの少なくとも一方の異常の有無を検査すべくコンピュータを通じて実行する手段として、

- a. 前記制御プログラムを自動生成する際に取得される情報から前記制御モデルと前記制御プログラムとの対応関係を表す対応情報を生成し、前記制御モデルをシミュレーションするシミュレーション手段と前記制御プログラムを実行するプログラム実行手段との双方の動作の停止箇所を前記対応情報に基づき関連付けする手段、
- b. 前記シミュレーション手段と前記プログラム実行手段との双方に対してシミュレーション及びプログラムの実行を指示する手段、
- c. 前記シミュレーション及びプログラムの実行の後、前記シミュレーション手段及び前記プログラム実行手段の双方の停止を検出する手段、
- d. 前記双方の停止を検出した後、前記シミュレーション手段によるシミュレーション結果及び前記プログラム実行手段による実行結果を出力する手段、

を含むことを特徴とする制御プログラムの検査プログラム。

【請求項 30】

前記シミュレーション手段によるシミュレーション結果及びプログラム実行手段による実行結果を出力する手段は、それら各動作結果を表示手段に出力して可視表示させるものである

請求項 2 9 記載の制御プログラムの検査プログラム。

【請求項 3 1】

前記対応情報は、前記制御モデルと前記制御プログラムとの互いに対応する位置の関係を表す情報である実行位置対応情報を含み、

前記対応情報に基づき停止箇所を関連付けする手段は、前記シミュレーション手段及び前記プログラム実行手段のいずれか一方においてその動作の停止箇所を指定するブレークポイントが設定されることによって前記実行位置対応情報に基づき他方の対応する箇所を指定するブレークポイントを設定するものである

請求項 2 9 又は 3 0 記載の制御プログラムの検査プログラム。

【書類名】明細書

【発明の名称】制御プログラムの検査方法及び検査装置及び検査プログラム

【技術分野】

【0001】

本発明は、制御対象の制御に用いる制御プログラムの検査方法及び該制御プログラムの検査に際して用いられる検査装置及び検査プログラムに関する。

【背景技術】

【0002】

例えば車載エンジン等を制御する制御プログラムの開発には通常、

(ア) 設計 (制御) 仕様 (制御モデル) に基づいて制御プログラムを作成 (コーディング) する工程。

(イ) 該作成した制御プログラムの異常の有無を検査、修正 (デバッグ) する工程。

(ウ) 該デバッグした制御プログラムを用いて実際の車両にてデバッグ、適合 (実車チェック) する工程。

(エ) この実車チェック結果に基づいて更に制御プログラムを修正する工程。  
等々が含まれる。

【0003】

また近年は、こうした制御プログラムの開発において、その要求仕様を記述する制御モデルからプログラム言語にて記述された制御プログラムを自動的に生成する工程を有するモデルベース開発も行われている。このモデルベース開発では、制御モデルをコンピュータ上で既存のソフトウェアによりシミュレーションすることで、その制御モデルの動作等の妥当性が検証される。こうした検証のなされた制御モデルから自動生成された制御プログラムは完成度が高いため、上記 (エ) の工程における制御プログラムの修正を低減することができるようになる。また、制御モデルの構築自体は、制御プログラムの作成よりも簡易に行うことができる。そのため、こうしたモデルベース開発を採用することで、制御プログラムの開発の効率化を図ることも可能となる。

【発明の開示】

【発明が解決しようとする課題】

【0004】

ところで、上記自動生成された制御プログラムには、制御モデルの指定した各処理について、同制御モデルでは必ずしも決められていなかった実行順序が、自動生成された制御プログラムでは新たに規定されることがある。

【0005】

このため、上記モデルベース開発を採用する場合、制御モデルから自動生成される制御プログラムのデバッグ作業や、上記制御モデルが制御プログラムを自動生成するのに妥当なものであるか否かを検査し修正するデバッグ作業が逆に困難なものとなる。これは、上述のように制御モデルと制御プログラムの実行順序が必ずしも同じではないことに起因するものであり、制御モデル及び制御プログラムの一方のデバッグ作業にて何らかの問題が発見されたときには、その問題の位置と対応する他方の位置を特定することが極めて困難となる。

【0006】

本発明は、こうした実情に鑑みてなされたものであり、その目的は、モデルベース開発を行う場合にあって、制御プログラムや制御モデルの異常の有無の検査を簡易に行うことのできる制御プログラムの検査方法、及び該制御プログラムの検査に用いる検査装置及び検査プログラムを提供することにある。

【課題を解決するための手段】

【0007】

こうした目的を達成すべく、請求項 1 記載の制御プログラムの検査方法あるいは請求項 13 記載の制御プログラムの検査装置では、前記制御モデルの挙動をシミュレーションするシミュレーション手段の動作結果と前記制御プログラムを実行するプログラム実行手段



の動作結果とのそれぞれ対応する動作結果毎にこれらを関連付けして出力して、前記制御モデル及び前記制御プログラムの少なくとも一方の異常の有無を検査するようにした。

#### 【0008】

このように、制御プログラムをプログラム実行手段によって実行することでその異常の有無の検査を行うに際し、同制御プログラムの実行結果と制御モデルのシミュレーション結果との各対応するものをそれぞれ関連付けて出力することで、これら制御プログラムや制御モデルの異常の有無の検査を簡易に行うことができるようになる。

#### 【0009】

また、請求項2記載の制御プログラムの検査方法あるいは請求項14記載の制御プログラムの検査装置では、前記関連付けが、前記自動コード生成手段による制御プログラムの自動生成に際して取得される前記制御モデルと該制御モデルから生成される制御プログラムとの対応関係を表す対応情報に基づいて行われるようにした。

#### 【0010】

制御モデルから制御プログラムを自動生成する自動コード生成手段は、同自動生成に際し制御モデルと制御プログラムとの対応関係を表す対応情報を有することとなる。そして、この対応情報を用いることで、シミュレーション手段による制御モデルのシミュレーションの実行位置とプログラム実行手段による制御プログラムの実行位置とを適切に関連付けることができる。

#### 【0011】

また、請求項3記載の制御プログラムの検査方法あるいは請求項15記載の制御プログラムの検査装置では、前記関連付けが、前記制御モデル及び前記制御プログラムのいずれか一方に対して動作の停止箇所を指定するブレークポイントが設定されることに基づき、他方に対しても前記対応情報に基づきその対応するブレークポイントを設定することによって行われるようにした。

#### 【0012】

シミュレーション手段やプログラム実行手段は、それぞれのブレークポイントでシミュレーションの実行や制御プログラムの実行を停止する。このため、対応情報により把握される制御モデルと制御プログラムとの対応関係に基づき、それぞれの対応する実行位置にブレークポイントを設定することで、制御モデル及び制御プログラムについての上記実行位置が互いに関連する位置となる。したがって、これらシミュレーション手段とプログラム実行手段との実行、停止を、既存の機能（ブレークポイント）を用いることで互いの同期（関連付け）を簡易に行うことができる。

#### 【0013】

更に、上記のように制御モデル及び制御プログラムのいずれか一方にブレークポイントが設定されたときに、他方にもブレークポイントが自動的に設定されるようにした。このため、例えば、外部からユーザが制御モデル及び制御プログラムのうちのいずれか一方を任意に選択してブレークポイントを設定することで、双方の適切な箇所にブレークポイントを設定することができる。

#### 【0014】

また、請求項4記載の制御プログラムの検査方法あるいは請求項16記載の制御プログラムの検査装置では、前記ブレークポイントを、前記制御モデルを構成する各機能ブロック毎に設定可能とした。

#### 【0015】

制御モデルから自動生成される制御プログラムには、制御モデルの各機能ブロックに対応する処理について同制御モデルには規定されていない実行順序が新たに規定されることがある。このため、自動生成された制御プログラムの検査においては、これら新たに規定される実行順序の異常の有無を検査することが望ましい。この点、ブレークポイントを各機能ブロックに設定可能とすることで、新たに実行順序が規定される場合であれ、その異常の有無を、制御モデルと対応させつつ簡易に検査することができるようになる。

#### 【0016】

また、請求項5記載の制御プログラムの検査方法あるいは請求項17記載の制御プログラムの検査装置では、前記各動作結果についてのそれぞれ関連付けされた動作結果の出力が、前記プログラム実行手段により前記制御プログラムを1行ずつ実行及び停止させることによって行われるようにした。

#### 【0017】

このように、制御プログラムを1行ずつ実行及び停止させるようにすることで、停止箇所の設定を簡易に行うことができ、より詳細な検査をすることができる。

また、請求項6記載の制御プログラムの検査方法あるいは請求項18記載の制御プログラムの検査装置では、前記異常の有無の検査が、前記関連付けされた前記シミュレーション手段の動作結果と前記プログラム実行手段の動作結果とを逐次比較する比較手段によって行われるようにした。

#### 【0018】

このように、制御モデルのシミュレーション結果と制御プログラムの実行結果とを逐次比較する比較手段を用いることで、これら制御モデルのシミュレーション結果と制御プログラムの実行結果との対応関係を、ユーザが確認する必要が無く、自動的に検出することができるようになる。したがって、上記異常の有無の検査を自動的に行うことができる。また、請求項5記載の方法や請求項17記載の構成と併用することによって、より細部までの実行位置の比較ができ、詳細な検査をすることができるようになる。

#### 【0019】

また、請求項7記載の制御プログラムの検査方法あるいは請求項19記載の制御プログラムの検査装置では、前記比較手段によるシミュレーション手段の動作結果と前記プログラム実行手段の動作結果との逐次比較が、前記対応情報に基づく前記制御モデルのシミュレーション順序と前記制御プログラムの実行順序との比較として行われるようにした。

#### 【0020】

自動生成される制御プログラムには、制御モデルの処理について同制御モデルが有しない実行順序が新たに規定されることがある。このため、同制御プログラムと制御モデルとは互いの処理の実行順序が異なったものとなることがある。この点、上記のように比較手段によって制御モデルのシミュレーションの順序と制御プログラムの実行順序とを比較することで、この実行順序の不一致にかかる異常を自動的に検出することができるようになる。

#### 【0021】

また、請求項8記載の制御プログラムの検査方法あるいは請求項20記載の制御プログラムの検査装置では、前記比較手段によるシミュレーション手段の動作結果と前記プログラム実行手段の動作結果との逐次比較が、前記対応情報に基づき前記制御モデルのシミュレーションによって演算される変数値と前記制御プログラムの実行によって演算される変数値との比較として行われるようにした。

#### 【0022】

自動生成された制御プログラムが適切なものでないと、制御プログラムの実行によって演算される変数値は、制御モデルの対応する処理によって演算される変数値と異なることがある。この点、上記のように比較手段によって制御モデルのシミュレーションによって演算される変数値と制御プログラムの実行によって演算される変数値とを比較することで、これらの不一致にかかる異常を自動的に検出することができるようになる。

#### 【0023】

なお、こうした変数値の比較による異常の有無の検査は、請求項9記載の制御プログラムの検査方法あるいは請求項21記載の制御プログラムの検査装置によるように、前記制御モデルのシミュレーションによって演算される変数値と前記制御プログラムの実行によって演算される変数値との差が許容範囲内にはいか否かによって行われるようにしてもよい。制御モデルのシミュレーションを行うシミュレーション手段や制御プログラムの実行を行うプログラム実行手段の演算精度に依存して、演算された変数値がわずかながら一致しない場合がある。しかしこのような場合、制御モデル及び制御プログラムは共に正常で

あり、変数値が不一致でも正常と判定すべきものである。そういった場合は、変数値の偏差が許容範囲内にはないか否かによって異常の有無の検査を行うとよい。これにより、異常の有無の検査をより適切に行うことができるようになる。

【0024】

また、請求項10記載の制御プログラムの検査方法あるいは請求項22記載の制御プログラムの検査装置では、前記比較手段により異常がある旨の判断がなされたとき、その時点での前記制御モデルのシミュレーション箇所と前記制御プログラムの実行箇所とを前記検査の結果として出力するようにした。

【0025】

このように、制御モデルのシミュレーション箇所と制御プログラムの実行箇所とを検出し、検査の結果として出力することで、外部のユーザ等が異常がある旨判断された箇所を簡易に把握することができるようになる。

【0026】

また、請求項11記載の制御プログラムの検査方法あるいは請求項23記載の制御プログラムの検査装置では、前記検査に伴って前記制御モデルのシミュレーション及び前記制御プログラムの実行が停止されるとき、前記制御モデルのシミュレーションによって演算される変数値の停止箇所において保持されている値及び前記制御プログラムの実行によって演算される変数値の停止箇所において保持されている値の少なくとも一方を変更可能とした。

【0027】

例えば前記ブレークポイントや何らかの異常の判断に起因して制御モデルのシミュレーションや制御プログラムの実行が停止されるとき、それら制御モデルのシミュレーション停止箇所において保持される変数値及び制御プログラムの実行停止箇所保持される変数値の少なくとも一方を変更可能とすることで、たとえ異常がある旨の判断がなされた場合であれ、再度制御モデルのシミュレーション条件と制御プログラムの実行条件とを等しくすることができる。このため、異常がある旨判断された箇所から異常の有無の検査を再開することが可能となる。

【0028】

また、請求項12記載の制御プログラムの検査方法あるいは請求項24記載の制御プログラムの検査装置では、少なくとも前記制御モデルの挙動をシミュレーションするシミュレーション手段の動作結果と前記制御プログラムを実行するプログラム実行手段の動作結果とを表示手段に出力して可視表示させるようにした。

【0029】

このように、シミュレーション手段の動作結果とプログラム実行手段の動作結果とを表示手段に出力して可視表示させることで、ユーザ自身が各動作結果を簡易に把握することができるようになる。

【0030】

また、請求項25記載の制御プログラムの検査プログラムでは、前記制御モデル及び前記制御プログラムの少なくとも一方の異常の有無を検査すべくコンピュータを通じて実行する手段として、a. 前記制御プログラムを自動生成する際に取得される情報から前記制御モデルと前記制御プログラムとの対応関係を表す対応情報を生成し、前記制御モデルをシミュレーションするシミュレーション手段と前記制御プログラムを実行するプログラム実行手段との双方の動作の停止箇所を前記対応情報に基づき関連付けする手段、b. 前記シミュレーション手段と前記プログラム実行手段との双方に対してシミュレーション及びプログラムの実行を指示する手段、c. 前記シミュレーション及びプログラムの実行の後、前記シミュレーション手段及び前記プログラム実行手段の双方の停止を検出する手段、d. 前記双方の停止を検出したとき、前記制御モデルのシミュレーションの結果と前記制御プログラムの実行の結果とを比較し、その比較結果に基づいて前記異常の有無を検査する手段、を含むようにした。

【0031】

このようにシミュレーション手段とプログラム実行手段との双方の停止箇所を関連付けることで、シミュレーション手段とプログラム実行手段とを同期させることができるようになる。そして、これらシミュレーション手段とプログラム実行手段とのそれぞれに対し、実行を指示するとともに、双方の停止を検出したときにシミュレーションの結果と制御プログラムの実行の結果とを比較することで、双方の停止位置において異常の有無を的確に検査することができるようになる。

#### 【0032】

また、請求項 2 6 記載の制御プログラムの検査プログラムでは、前記対応情報は、前記制御モデルと前記制御プログラムとの互いに対応する位置の関係を表す情報である実行位置対応情報を含み、前記対応情報に基づき停止箇所を関連付けする手段が、前記シミュレーション手段及び前記プログラム実行手段のいずれか一方においてその動作の停止箇所を指定するブレークポイントが設定されることによって前記実行位置対応情報に基づき他方の対応する箇所を指定するブレークポイントを設定するものであるとした。

#### 【0033】

このような検査プログラムによれば、外部からユーザが制御モデル及び制御プログラムのうちのいずれか一方を任意に選択してブレークポイントを設定することに基づいて、双方の適切な箇所にブレークポイントが自動的に設定されるようになる。

#### 【0034】

また、請求項 2 7 記載の制御プログラムの検査プログラムでは、前記比較結果に基づいて異常の有無を検査する手段を、少なくとも前記制御モデルのシミュレーションの停止位置と前記制御プログラムの実行の停止位置とを前記実行位置対応情報に基づき比較することによって前記異常の有無を検査するものとした。

#### 【0035】

自動生成される制御プログラムには、制御モデルの処理についての実行順序であって同制御モデルが有しない実行順序が新たに規定されることがある。このため、同制御モデルを用いて作成される制御プログラムと制御モデルとは互いの処理の実行順序が異なったものとなることがある。この点、上記のように制御モデルの実行順序と制御プログラムの実行順序とを比較することで、この実行順序の不一致にかかる異常を適切に検出することができるようになる。

#### 【0036】

また、請求項 2 8 記載の制御プログラムの検査プログラムでは、前記対応情報は、前記制御モデルの処理にかかる変数値と前記制御プログラムの処理にかかる変数値との対応関係を表す変数対応情報を含み、前記比較結果に基づいて異常の有無を検査する手段を、少なくとも前記制御モデルのシミュレーションによって演算される変数値と前記制御プログラムの実行によって演算される変数値とを前記変数対応情報に基づき比較することによって前記異常の有無を検査するものとした。

#### 【0037】

自動生成された制御プログラムが適切なものでないと、制御プログラムの実行によって演算される変数値は、制御モデルの対応する処理によって演算される変数値と異なることがある。この点、制御モデルのシミュレーションによって演算される変数値と制御プログラムの実行によって演算される変数値とを比較することで、これら変数値の不一致にかかる異常についてもこれを的確に検出することができるようになる。

#### 【0038】

また、請求項 2 9 記載の制御プログラムの検査プログラムでは、前記制御モデル及び前記制御プログラムの少なくとも一方の異常の有無を検査すべくコンピュータを通じて実行する手段として、a. 前記制御プログラムを自動生成する際に取得される情報から前記制御モデルと前記制御プログラムとの対応関係を表す対応情報を生成し、前記制御モデルをシミュレーションするシミュレーション手段と前記制御プログラムを実行するプログラム実行手段との双方の動作の停止箇所を前記対応情報に基づき関連付けする手段、b. 前記シミュレーション手段と前記プログラム実行手段との双方に対してシミュレーション及び

プログラムの実行を指示する手段、c. 前記シミュレーション及びプログラムの実行の後、前記シミュレーション手段及び前記プログラム実行手段の双方の停止を検出する手段、d. 前記双方の停止を検出した後、前記シミュレーション手段によるシミュレーション結果及び前記プログラム実行手段による実行結果を表示手段に出力する手段、を含むようにした。

#### 【0039】

このように、シミュレーション手段とプログラム実行手段との双方の停止箇所を関連付けることで、シミュレーション手段とプログラム実行手段との各動作結果を同期させることができるようになる。そして、これらシミュレーション手段とプログラム実行手段とのそれぞれに対し、実行を指示するとともに、双方の停止を検出した後に、シミュレーション手段のシミュレーション結果及びプログラム実行手段の実行結果を出力することで、制御モデルの動作と制御プログラムの動作とを簡易に把握することができるようになる。

#### 【0040】

またこの場合、請求項30記載の制御プログラムの検査プログラムのように、前記シミュレーション手段によるシミュレーション結果及びプログラム実行手段による実行結果を出力する手段を、それら各動作結果を表示手段に出力して可視表示させるものとすることで、ユーザ自身がそれら動作結果を比較して制御プログラムや制御モデルの異常の有無の検査を行うことも可能となる。

#### 【0041】

また、請求項31記載の制御プログラムの検査プログラムでは、前記対応情報は、前記制御モデルと前記制御プログラムとの互いに対応する位置の関係を表す情報である実行位置対応情報を含み、前記対応情報に基づき停止箇所を関連付けする手段を、前記シミュレーション手段及び前記プログラム実行手段のいずれか一方においてその動作の停止箇所を指定するブレークポイントが設定されることによって前記実行位置対応情報に基づき他方の対応する箇所を指定するブレークポイントを設定するものとした。

#### 【0042】

このような検査プログラムによれば、外部からユーザが制御モデル及び制御プログラムのうちのいずれか一方を任意に選択してブレークポイントを設定することに基づいて、双方の適切な箇所にブレークポイントが自動的に設定されるようになる。

【発明を実施するための最良の形態】

#### 【0043】

以下、本発明にかかる制御プログラムの検査方法及び検査装置を車両の制御プログラムの検査に適用した一実施形態を図面を参照しつつ説明する。

図1は、本実施形態にかかる制御プログラムの作成並びに検査手順の概要を示すフローチャートである。

#### 【0044】

同図1に示すステップS1では、まず、対象となる制御プログラムについて、その要求仕様を記述する制御モデルを要求仕様記述言語を用いて構築する。次に、こうして構築された制御モデルをシミュレーションすることで、同制御モデルが正当な機能と妥当な性能を有するものであるか否かを検証する。そして、この検証結果に基づき、適宜、制御モデルを修正する。

#### 【0045】

次に、ステップS2において、制御モデルから例えばC言語にて記述された制御プログラムを自動生成する。この制御プログラムには、制御モデルの規定する処理について同制御モデルによっては規定されていない実行順序が新たに規定されている。

#### 【0046】

ステップS3では、上記制御モデルをシミュレーションするとともに、制御プログラムを実行することで、互いの対応関係を維持しつつ制御モデルや制御プログラムの異常の有無の検査を行う。そして、制御プログラムに異常がある旨判断されると、制御プログラムが修正される。また、制御プログラムを検証した結果、制御モデルが妥当でないと判断さ

れる場合には、上記ステップ S 1 に戻り、制御モデルを修正するようにしてもよい。

【0047】

こうして制御プログラムのデバッグが終了すると、ステップ S 4 において、同制御プログラムを搭載した電子制御装置によって実際の車両でのデバッグ、適合を行う。そして、この実際の車両でのデバッグ結果に基づき制御プログラムに妥当でない箇所がある場合には、ステップ S 1 に戻って制御モデルを修正し、上記ステップ S 1 ～ S 4 の処理を繰り返す。この一連の処理は、制御プログラムが妥当であると判断されるまで行われる。

【0048】

以下、上記ステップ S 3 の処理である制御プログラムのデバッグ作業について、詳細に説明する。

図 2 は、制御プログラムの生成及びデバッグに用いるシステム（検査装置）の全体構成を示す図である。

【0049】

このシステムにおいて、モデル格納部 10 には、先の図 1 のステップ S 1 にて構築され検証のなされた制御モデルが格納されている。図 3 に、こうした制御モデルの一例を模式的に示す。

【0050】

同図 3 に示す制御モデルは、車載エンジンの燃料噴射量の演算、及び噴射タイミングの演算を行うモデルである。ブロック B 1 は、図示しないエンジンのクランク角度が所定のクランク角と一致したときに、ブロック B 2 及びブロック B 6 にトリガを与えるブロックである。ブロック B 2 は、上記所定のクランク角をトリガとして基本噴射量を演算するブロックである。ブロック B 3 は、上記ブロック B 2 によって演算された基本噴射量をブロック B 4 に書き込み指示するブロックである。ブロック B 4 は、基本噴射量を記憶するブロックである。ブロック B 5 は、記憶されている基本噴射量を読み出すブロックである。ブロック B 6 は、上記所定のクランク角をトリガとして上記基本燃料噴射量に対する補正量を計算するブロックである。ブロック B 7 は、補正量と上記読み出された基本噴射量とに基づき噴射タイミングを演算するブロックである。

【0051】

ちなみに、図 3 において実線で示す矢印は、処理の順序を規定するとともに、各処理の施されたデータの流れを示すものである。すなわち、例えばブロック B 7 の処理は、ブロック B 5 及びブロック B 6 の処理の後に行われるようにその処理順序が規定されている。しかし、ブロック B 2 とブロック B 6 との処理については、どちらを先に実施するかといった順序は規定されていない。また、ブロック B 3 及びブロック B 5 の処理の順序についても同様に、基本噴射量のメモリへの書き込みとメモリからの読み出しの順序は規定されていない。つまり、ブロック B 5 を先に処理し、ブロック B 3 を後で処理することも、このモデルでは考えられる。

【0052】

一方、図 2 に示すシステムにおいて、自動コード生成部（自動コード生成手段）12 は、この制御モデルを入力し、同制御モデルから上記 C 言語で記述された制御プログラムを生成してプログラム格納部 14 へ格納する部分である。こうして、プログラム格納部 14 には、上記自動コード生成部 12 により生成された制御プログラムが格納される。

【0053】

図 4 に、先の図 3 に示した制御モデルから自動生成される制御プログラムを例示する。同図 4 に示すように、先の図 3 に示した制御モデルにおけるブロック B 3 の処理はブロック B 5 の処理に先立って行われる必要があるにもかかわらず、この制御プログラムでは、ブロック B 5 に対応する処理の後にブロック B 3 に対応する処理が行われる。これは、制御モデルには各機能ブロックの示す処理の実行順序が必ずしも定められているわけではなく、自動コード生成部 12 によってこの実行順序が新たに規定されることに起因している。すなわち上述のように、制御モデルにはブロック B 5 とブロック B 3 との処理の順序が規定されていないため、これらの処理の実行順序が自動コード生成部 12 によって新たに

規定される際、適切なものとならなかった場合を図4は示している。

【0054】

また、図2に示すシステムにおいて、対応情報作成部16は、上記自動コード生成部12が制御プログラムを生成するに際して、同自動コード生成部12の有する情報に基づいて制御モデルと制御プログラムとの対応関係を表す対応情報を作成し、これを対応情報格納部18へ格納する部分である。この対応情報は、制御モデルのブロックと制御プログラムの実行ポイントとの対応関係を表す実行位置対応情報と、制御モデルのブロック結線（ブロック間の結線が表す変数）やストレージ（ブロックB4等、記憶にかかるブロック）と制御プログラムの変数との対応関係を表す変数対応情報とを有している。

【0055】

図5に、これら実行位置対応情報と変数対応情報とからなる対応情報を例示する。同図5においては、実行位置対応情報として、制御モデルのブロックと、制御プログラムの有するラベルL1～L5とが対応付けられている。また、変数対応情報として、制御モデルのブロック結線／ストレージと制御プログラムの変数とが対応付けられている。これら対応情報が、図2に示した対応情報格納部18に格納されることになる。

【0056】

また、図2に示すシステムにおいて、シミュレーション部（シミュレーション手段）20は、モデル格納部10から制御モデルを入力し、その動作をシミュレーションする部分である。

【0057】

他方、同図2に示すシステムにおいて、プログラム実行部（プログラム実行手段）22は、プログラム格納部14から制御プログラムを入力し、これを実行する部分である。なお、この実施形態においては、このプログラム実行部22としてデバッガ機能付きのコンパイラを利用している。

【0058】

また、同システムにおいて、ブレークポイント設定部24は、外部からの入力に基づき制御モデル及び制御プログラムのいずれかに動作の停止箇所を指定するブレークポイントの設定をシミュレーション部20及びプログラム実行部22に指示する部分である。このブレークポイントは、制御モデルの各ブロック毎に設定可能とする。また、このブレークポイントは、例えば制御プログラムの上記ラベルL1～L5を用いて指定される各実行ポイントに設定可能とする。

【0059】

また、図2に示すシステムにおいて、同期部（同期手段）26は、上記ブレークポイント設定部24を通じて制御モデル及び制御プログラムのいずれか一方に、その動作の停止箇所を指定するブレークポイントが設定されたとき、上記対応情報に基づき、他方に対応する動作の停止箇所を指定するブレークポイントを設定する機能を有する部分である。すなわちここでは、上記対応情報のうちの実行位置対応情報を用いて、制御モデルのブロックから対応する制御プログラムの実行ポイントを検索することで、又は、制御プログラムの実行ポイントから制御モデルのブロックを検索することで、ブレークポイントを設定する。

【0060】

更に、この同期部26は、シミュレーション部20及びプログラム実行部22の動作が双方ともブレークポイントにて一旦停止したときに、これらシミュレーション部20及びプログラム実行部22の動作結果を比較する比較部（比較手段）を備えている。

【0061】

また、図2に示すシステムにおいて、表示部（表示手段）28は、通常の場合案内表示等をはじめ、上記比較部による比較結果により不一致が検出され、制御モデル及び制御プログラムのいずれかに問題があるときなどにこれを可視表示する部分である。また、結果格納部30には、上記比較部による比較結果が格納される。

【0062】



ちなみに、この図2に示すシステムにおける上記モデル格納部10、プログラム格納部14、対応情報格納部18、結果格納部30は、例えばハードディスク装置等の記憶装置によって構成される。また、自動コード生成部12、対応情報作成部16、シミュレーション部20、プログラム実行部22、ブレークポイント設定部24、同期部26は、処理手順に関するプログラムが記録された例えばハードディスク装置やROM等の記憶装置とコンピュータとによって構成される。

#### 【0063】

以下、こうした構成を有するシステム（検査装置）を通じて実行される制御プログラムの検査及びデバッグにかかる処理について詳細に説明する。

本実施形態では、モデル格納部10に格納された制御モデルをシミュレーションするシミュレーション部20とプログラム格納部14に格納された制御プログラムを実行するプログラム実行部22との各動作結果を同期部26を通じて同期（関連付け）させる。そして、上述したブレークポイントでこれらシミュレーション部20とプログラム実行部22との動作が停止したとき、上記比較部によってシミュレーションの結果とプログラムの実行結果とを比較することで、制御モデルや制御プログラムの異常の有無を検査する。そして、この検査の結果、制御プログラムに問題があるときには、又は、制御モデルが妥当な制御プログラムを生成することができないときには、制御プログラムや制御モデルを修正する。

#### 【0064】

このように、制御プログラムの実行結果と制御モデルのシミュレーション結果とを関連付けて比較することで、制御モデルが規定する各処理について同制御モデルの有しない実行順序が新たに規定されることがある制御プログラムと、制御モデルとの対応関係を簡易に把握することができるようになる。したがって、モデルベース開発を行う場合にあって、制御プログラムや制御モデルの異常の有無の検査を簡易に行うことができるようになる。

#### 【0065】

ここでまず、上記ブレークポイント設定部24及び同期部26によるブレークポイントの設定にかかる処理について図6を参照して説明する。なお、同図6は、同期部26によるブレークポイントの設定にかかる処理の手順を示すフローチャートである。

#### 【0066】

この一連の処理においては、ステップS301及びステップS302において、上記ブレークポイント設定部24を通じて上記シミュレーション部20及び上記プログラム実行部22のいずれかからブレークポイントが設定されたか否かを判断する。すなわち、同期部26では、シミュレーション部20及びプログラム実行部22にアクセスすることで、ブレークポイントの設定がなされたか否かを判断する。

#### 【0067】

そして、ステップS301においてブレークポイントの設定がなされた旨判断されると、ステップS303の処理に移行する。このステップS303では、実行位置対応情報に基づいて制御モデル上に設定されたブレークポイントに該当する制御プログラム上の位置を決定するとともに、この決定された位置をブレークポイントとして制御プログラム上に設定するようにプログラム実行部22に指示する。

#### 【0068】

また、ステップS302においてブレークポイントの設定がなされた旨判断されると、ステップS304の処理に移行する。このステップS304では、実行位置対応情報に基づいて制御プログラム上に設定されたブレークポイントに該当する制御モデル上の位置を決定するとともに、この決定された位置をブレークポイントとして制御モデル上に設定するようにシミュレーション部20に指示する。

#### 【0069】

そして、ステップS303、S304の処理が終了した場合やステップS302にてブレークポイントの設定がなされていない旨判断された場合には、この一連の処理を終了す



る。なお、この一連の処理を、所定周期で繰り返し実行する代わりに、シミュレーション部 20 に制御モデルが取り込まれるとともに、プログラム実行部 22 に制御プログラムが取り込まれることをトリガとして行うようにすることがより望ましい。

#### 【0070】

ちなみに、このブレークポイントは、先の図 3 に示した制御モデルにおいては、例えばブロック B3 とブロック B5 とにそれぞれ設けるようにすればよい。また、最終的な出力となるブロック B7 にのみ設けるようにしてもよい。

#### 【0071】

次に、上記シミュレーション部 20 による制御モデルのシミュレーションにかかる処理について図 7 を用いて説明する。なお、同図 7 は、制御モデルのシミュレーションにかかる処理の手順を示すフローチャートである。

#### 【0072】

この一連の処理においては、まずステップ S311 において、モデル格納部 10 に格納されている制御モデルを取り込む。次に、ステップ S312、S313 において、制御モデルの所定箇所にブレークポイントを設定する。すなわち、このステップ S312、S313 においては、上記ブレークポイント設定部 24 の指示によりブレークポイントを設定するか、先の図 6 に示した処理にて同期部 26 の指示によりブレークポイントを設定するかのいずれかの処理を行う。

#### 【0073】

続くステップ S314 においては、上記同期部 26 からシミュレーションの実行の指示があったか否かを判断する。そして、ステップ S314 にて実行の指示があった旨判断されると、ステップ S315 にて制御モデルのシミュレーションを実行する。このシミュレーションは、上記シミュレーション部 20 に例えばセンサ信号等に相当する擬似的なデータを与え、同シミュレーション部 20 に上記制御モデルに基づきそのデータを処理させるようにして行われる。

#### 【0074】

また、このシミュレーションは、制御モデルの全ての実行が終了するまで（ステップ S316）、又は、ブレークポイントを検出するまで（ステップ S317）行われる。そして、ステップ S317 にてブレークポイントが検出されると、シミュレーションを停止させるとともに、ステップ S318 において、上記表示部 28、あるいはシミュレーション部 20 自身の備える表示器等の出力装置を通じて外部にシミュレーションを停止している旨を通知する。

#### 【0075】

このようにブレークポイントにてシミュレーションを停止すると、上記同期部 26 からの新たな実行指示（ステップ S314）、又は同期部 26 からの異常通知（ステップ S319）に備えて待機することとなる。そして、新たな実行指示があった場合には、ステップ S315 以降の処理を行うことで、制御モデルのシミュレーションを再開する。

#### 【0076】

一方、ステップ S319 において同期部 26 からの異常通知があった場合にはこの一連の処理を終了する。なお、この際、シミュレーション部 20 は異常通知があったときの停止位置の情報を有しているため、これを上記表示部 28 や同シミュレーション部 20 自身の備える表示器等の出力装置を通じて外部に出力することが望ましい。さらに、同期部 26 から異常通知があった場合には、上記結果格納部 30（図 2）に異常情報を格納し、ステップ S314 にて、同期部 26 からの実行指示に備えて待機するように構成してもよい（図 7 の破線参照）。また、この異常通知が変数についての異常通知である場合、当該シミュレーションによって演算された変数値を変更可能としてもよい。すなわち、再度制御モデルのシミュレーション条件と制御プログラムの実行条件とを等しくすべく同シミュレーションによって演算された変数値を変更してステップ S314 にて待機するようにしてもよい。

#### 【0077】

また、ステップ S 3 1 6 において、全てのシミュレーションが完了した旨判断される場合には、ステップ S 3 2 0 において、上記表示部 2 8 やシミュレーション部 2 0 自身の備える表示器等の出力装置を通じて外部にシミュレーションが完了した旨を通知し、この一連の処理を終了する。

#### 【0078】

次に、上記プログラム実行部 2 2 による制御プログラムの実行にかかる処理について図 8 を用いて説明する。なお、同図 8 は、制御プログラムの実行にかかる処理の手順を示すフローチャートである。

#### 【0079】

この一連の処理においては、まずステップ S 3 3 1 において、プログラム格納部 1 4 に格納されている制御プログラムを取り込む。次に、ステップ S 3 3 2、S 3 3 3 において、制御プログラムの所定箇所にブレークポイントを設定する。すなわち、このステップ S 3 3 2、S 3 3 3 においては、上記ブレークポイント設定部 2 4 の指示によりブレークポイントを設定するか、先の図 6 に示した処理にて同期部 2 6 の指示によりブレークポイントを設定するかのいずれかの処理を行う。

#### 【0080】

続くステップ S 3 3 4 においては、上記同期部 2 6 から制御プログラムの実行の指示があったか否かを判断する。そして、ステップ S 3 3 4 にて実行の指示があった旨判断されると、ステップ S 3 3 5 にて制御プログラムを実行する。この制御プログラムの実行も、上記プログラム実行部 2 2 に例えばセンサ信号等に相当する擬似的なデータを与え、同プログラム実行部 2 2 に上記制御プログラムに基づきそのデータを処理させるようにして行われる。

#### 【0081】

また、このプログラムの実行は、制御プログラムの全てを実行するまで（ステップ S 3 3 6）、又は、ブレークポイントを検出するまで（ステップ S 3 3 7）行われる。そして、ステップ S 3 3 7 にてブレークポイントが検出されると、制御プログラムの実行を停止させるとともに、ステップ S 3 3 8 において、上記表示部 2 8 やプログラム実行部 2 2 自身の備える表示器等の出力装置を通じて外部に制御プログラムの実行を停止している旨を通知する。

#### 【0082】

このようにブレークポイントにて制御プログラムの実行を停止すると、上記同期部 2 6 からの新たな実行指示（ステップ S 3 3 4）、又は同期部 2 6 からの異常通知（ステップ S 3 3 9）に備えて待機することとなる。そして、新たな実行指示があった場合には、ステップ S 3 3 5 以降の処理を行うことで、制御プログラムの実行を再開する。

#### 【0083】

一方、ステップ S 3 3 9 において同期部 2 6 からの異常通知があった場合にはこの一連の処理を終了する。なお、この際、プログラム実行部 2 2 は異常通知のあったときの停止位置の情報を有しているため、これを上記表示部 2 8 や同プログラム実行部 2 2 自身の備える表示器等の出力装置を通じて外部に出力することが望ましい。さらに、同期部 2 6 から異常通知があった場合には、上記結果格納部 3 0（図 2）に異常情報を格納し、ステップ S 3 3 4 にて同期部 2 6 からの実行指示に備えて待機するように構成してもよい（図 8 の破線参照）。また、異常通知が変数についての異常通知である場合、当該制御プログラムの実行によって演算された変数値を変更可能としてもよい。すなわち、再度制御モデルのシミュレーション条件と制御プログラムの実行条件とを等しくすべく同制御プログラムの実行によって演算された変数値を変更してステップ S 3 3 4 にて待機するようにしてもよい。

#### 【0084】

また、ステップ S 3 3 6 において、全ての制御プログラムの実行が完了した旨判断される場合には、ステップ S 3 4 0 において、上記表示部 2 8 やプログラム実行部 2 2 自身の備える表示器等の出力装置を通じて外部に制御プログラムの実行が完了した旨を通知し、

この一連の処理を終了する。

【0085】

次に、上記同期部 26 により行われる制御モデル及び制御プログラムの異常の有無の検査にかかる処理について図 9 を用いて説明する。なお、同図 9 は、この検査にかかる処理の手順を示すフローチャートである。この処理は、先の図 6 に示した処理が終了したことを前提として、換言すれば、制御モデル及び制御プログラムの双方にブレークポイントが設定されたことを前提として行われる。

【0086】

この一連の処理においては、まずステップ S 351 において、制御モデルのシミュレーションを実行するように上記シミュレーション部 20 に指示するとともに、制御プログラムを実行するように上記プログラム実行部 22 に指示する。これら制御モデルのシミュレーションと制御プログラムの実行は、上記シミュレーション部 20 により制御モデルが停止されて且つ、上記プログラム実行部 22 により制御プログラムが停止されるまでの間にわたって行われる。なおこの際、制御モデルのシミュレーションと制御プログラムの実行とを同一の条件で行うべく、制御モデル及び制御プログラムへの入力データ（初期値）は同一データとする。

【0087】

そして、ステップ S 352 において、制御モデル及び制御プログラムの双方が停止されていると判断され、且つステップ S 353 においてこれら制御モデル及び制御プログラムの双方が実行完了していないと判断されると、ステップ S 354 以降の処理において、上記制御モデル及び制御プログラムの異常の有無についての検査を行う。

【0088】

すなわち、ステップ S 354 では、上記実行位置対応情報に基づき、制御モデルの停止位置と、制御プログラムの停止位置とが互いに対応する位置となっているか否かを確認する。そして、互いに対応する位置となっていないと判断されると（ステップ S 355）、ステップ S 356 において上記表示部 28 を通じて制御モデルと制御プログラムとの停止位置が互いに一致しない旨（「停止位置異常」）を表示する。

【0089】

一方、ステップ S 355 において停止位置が一致すると判断されると、ステップ S 357 の処理に移行する。このステップ S 357 では、上記変数対応情報に基づき、制御モデル側の当該停止箇所のブロック結線やストレージの値（変数値）と、制御プログラム側の当該停止箇所において保持されている変数値とが互いに対応するものであるか否かを判断する。そして、互いに対応するものでないとは判断されると（ステップ S 358）、ステップ S 359 において上記表示部 28 を通じて制御モデルのブロック結線やストレージと制御プログラムの変数値とが互いに対応するものでない旨（「変数値異常」）を表示する。

【0090】

ちなみに、上記制御モデルのブロック結線やストレージの値と制御プログラムの変数値とが互いに対応するものであるか否かは、これら 2 つの値の差としての許容範囲を予め定めておき、この許容範囲にあるか否かで判断するようにする。例えば、制御モデルのブロック結線やストレージの値と制御プログラムの変数値とが互いに整数演算された値である場合には、両者の一致のみを許容範囲とする。また例えば、制御モデルのブロック結線やストレージの値と制御プログラムの変数値とが互いに浮動小数点型の値である場合には、両者の差の絶対値が所定値以下であることを許容範囲とする。

【0091】

そして、上記ステップ S 358 において、変数値が許容範囲にあると判断されたときには、ステップ S 351 に戻り、制御モデルのシミュレーションと制御プログラムの実行とが再開されることとなる。また、ステップ S 353 において、制御モデルと制御プログラムとが全て実行されたと判断されると、ステップ S 360 において、上記表示部 28 を通じてその旨（「全て実行完了」）を表示する。

【0092】

一方、上記ステップ S 3 5 6、S 3 5 9、S 3 6 0 の各処理が終了すると、この一連の処理を終了する。なお、上記ステップ S 3 5 3 において制御モデルと制御プログラムとが全て実行されたと判断されたときには、ステップ S 3 6 0 の処理に移行するに先立って、ステップ S 3 5 7 等の処理を行うようにしてもよい。

#### 【0093】

このように本実施形態では、上記同期部 2 6（比較部）により制御モデルの停止位置と、制御プログラムの停止位置とが互いに対応する位置となっているか否かを判断するようにした。これにより、制御モデルの指定する複数の処理について、上記自動コード生成部 1 2 によって生成された制御プログラムに新たに実行順序が規定されたとき、これが適切でない場合にはこれを検出することができる。

#### 【0094】

また、処理順序に誤りがある場合や生成された制御プログラム自体に誤りがある場合には、図 9 の処理により、変数値の違いを検出することができる。

このように、本実施形態では、制御プログラムのデバッグ作業において問題が検出されたとき、制御モデルの位置及び同制御モデルでの処理状態（ブロック結線やストレージ）との関連付けを自動的に行うことができる。

#### 【0095】

そして、以上説明した作業、すなわち先の図 1 に示したステップ S 3 の処理にあたる制御プログラムのデバッグを行った結果、制御プログラムが妥当である旨判断されれば、次に、同制御プログラムを例えば ROM（Read Only Memory）等に焼き付けて（記憶させて）、これを電子制御装置に搭載することとなる。その後、ステップ S 4（図 1）の処理、すなわち同制御プログラムを実装した電子制御装置によって実際の車両でのデバッグ、適合を行い、制御プログラムに異常がある旨判断されるときは、制御プログラムや制御モデルの修正を検討することとなるのは前述した通りである。

#### 【0096】

以上詳述したように、本実施形態によれば、以下の効果が得られるようになる。

（1）制御プログラムをプログラム実行部 2 2 によって実行することでその異常の有無の検査を行うに際し、これを制御モデルのシミュレーションと同期させるようにすることで、これら制御モデルのシミュレーション状況と制御プログラムの実行状況とを簡易に比較することができるようになる。

#### 【0097】

（2）対応情報作成部 1 6 の作成する対応情報を用いることで、シミュレーション部 2 0 による制御モデルのシミュレーションの実行位置とプログラム実行部 2 2 による制御プログラムの実行位置とを適切に対応付けることができる。

#### 【0098】

（3）制御モデル及び制御プログラムのいずれか一方にブレークポイントが設定されたときに、同期部 2 6 により他方にもブレークポイントが自動的に設定されるようにした。このため、例えば、外部からユーザが制御モデル及び制御プログラムのうちのいずれか一方を任意に選択してブレークポイントを設定することで上記同期処理が可能となる。

#### 【0099】

（4）ブレークポイントを制御モデルの各機能ブロック毎に設定可能としたことで、生成される制御プログラムに新たに実行順序が規定される場合であれ、その異常の有無を、制御モデルと対応させつつ簡易に検査することができるようになる。

#### 【0100】

（5）制御モデルのシミュレーションの順序と制御プログラムのプログラム実行順序とを比較することで、この実行順序の不一致にかかる異常を自動的に検出することができるようになる。

#### 【0101】

（6）比較部によって制御モデルのシミュレーションによって生成される値と制御プログラムの実行にかかる変数値とを比較することで、これらの不一致にかかる異常を自動的

に検出することができるようになる。

【0102】

(7) 変数についての異常通知があったとき、図7のステップS319及び図8のステップS339においてそれら変数値の少なくとも一方を変更可能とした場合には、次のような効果も奏される。すなわち、変数異常がある旨判断がなされた場合であれ、再度制御モデルのシミュレーション条件と制御プログラムの実行条件とを等しくすることができ、ひいては異常がある旨判断された箇所から異常の有無の検査を再開することが可能となる。

【0103】

(8) 比較部による比較結果が不一致のときにこれを表示部28に出力して可視表示させることで、ユーザ自身が異常がある旨判断された箇所を簡易に把握することができるようになる。

【0104】

なお、上記実施形態は、以下のように変更して実施してもよい。

- ・シミュレーション部20及びプログラム実行部22の動作の停止箇所は、上述したものに限らない。例えば制御プログラムをプログラム実行部22により1行ずつ実行及び停止させつつシミュレーション部20をこれと同期させてもよい。これは、例えば制御プログラム側に1行ずつブレークポイントを設定するとともに、先の図6に示した処理によって制御モデルのブレークポイントを設定することで行ってもよい。

【0105】

- ・シミュレーション部20やプログラム実行部22は、必ずしも制御モデルや制御プログラムに入力データを与えることでシミュレーション及びプログラムの実行を行うものに限らない。例えば制御プログラムの制御対象及び同対象を取り巻く環境をコンピュータ上で擬似的に生成しつつ上記制御モデルや制御プログラムの検査を行うものであってもよい。

【0106】

- ・ブレークポイント設定部24の機能についてこれを上記シミュレーション部20自身に備えるようにしてもよい。また、同機能を上記プログラム実行部22自身に備えるようにしてもよい。

【0107】

- ・上記実施形態では、異常の有無の検査に際して用いられる許容範囲を変数値のデータ型に応じて変更するようにしたが、許容範囲の設定の仕方は任意である。また、必ずしも許容範囲を設定する必要はなく、両者の一致のみを異常の有無の判断基準としてもよい。

【0108】

- ・上記実施形態では、変数についての異常通知があった場合に、図7のステップS319及び図8のステップS339においてそれら変数値の少なくとも一方を変更可能とすることについても考慮した。これに代えて、例えば正常時に上記変数値としてあえて異常データを入力して制御プログラムの検査を行うように構成してもよい。

【0109】

- ・制御モデルから自動生成される制御プログラムとしては、C言語にて記述されるものに限らず、適宜のプログラム言語にて記述される制御プログラムであればよい。この制御プログラムが制御モデルの規定する処理について同制御モデルにおいて規定されない実行順序が新たに規定されることがあるものであるなら、この制御プログラムと制御モデルとを同期させて検査をすることは特に有効である。

【0110】

- ・上記実施形態では、同期部26が、制御モデルのシミュレーション結果と制御プログラムの実行結果とを比較する比較部を備え、比較結果が不一致のときにこれを表示部28に出力する構成とした。これに代えて、同期部26が、制御モデルのシミュレーション結果と制御プログラムの実行結果を受信して比較することなく、そのまま表示部28に出力する構成としてもよい。また、表示部28への出力は同期部26を経由せず、直接シミュ

レーション部 2 0 及びプログラム実行部 2 2 から表示部 2 8 にて表示してもよい。また、上記同期部 2 6 を経由して、あるいは経由せずに出力される各動作結果を、データファイル等として適宜の記憶装置に格納する構成としてもよい。

【0 1 1 1】

・上記対応情報作成部 1 6 や同期部 2 6 等をコンピュータとソフトウェアにて構成する代わりに、専用のハードウェアにて構成してもよい。

・制御プログラムの検査工程としては、モデルベース開発を用いるとともに、制御モデルのシミュレーションと制御プログラムの実行とを同期させつつこれらの異常の有無を検査する範囲で適宜変更してよい。

【0 1 1 2】

・車載エンジンの制御プログラムの作成に限らず、モデルベース開発を採用して制御プログラムを作成する場合には、本発明の適用は有効である。

【図面の簡単な説明】

【0 1 1 3】

【図 1】 本実施形態にかかる制御プログラムの作成並びに検査手順を示すフローチャート。

【図 2】 同実施形態にかかる制御プログラムの生成及び検査を行うシステムの構成を示すブロック図。

【図 3】 制御モデルを例示するブロック図。

【図 4】 自動生成される制御プログラムを例示する図。

【図 5】 上記実施形態にかかる制御モデルと制御プログラムとの対応関係を表す対応情報を例示する図。

【図 6】 同実施形態におけるブレイクポイントの設定にかかる処理の手順を示すフローチャート。

【図 7】 同実施形態にかかるシミュレーション部による処理の手順を示すフローチャート。

【図 8】 同実施形態にかかるプログラム実行部による処理の手順を示すフローチャート。

【図 9】 同実施形態にかかる同期部による処理の手順を示すフローチャート。

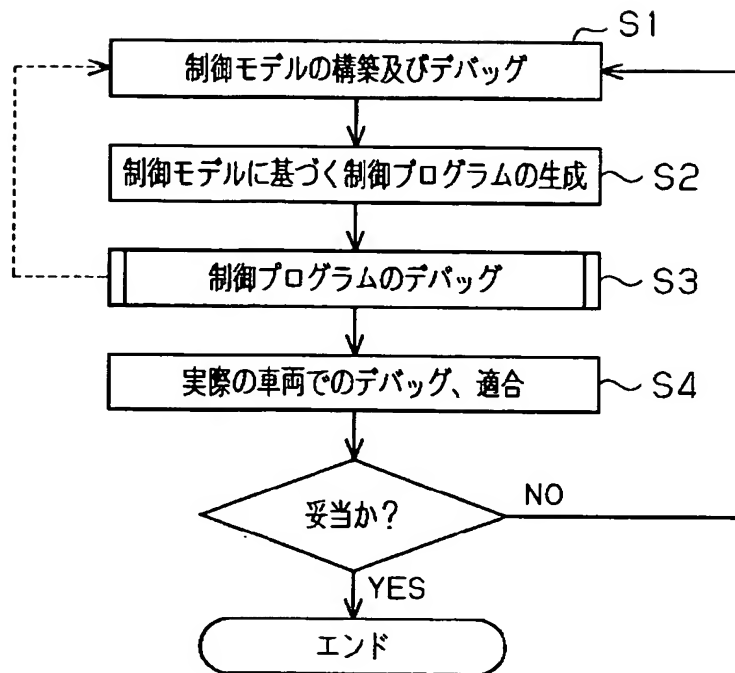
【符号の説明】

【0 1 1 4】

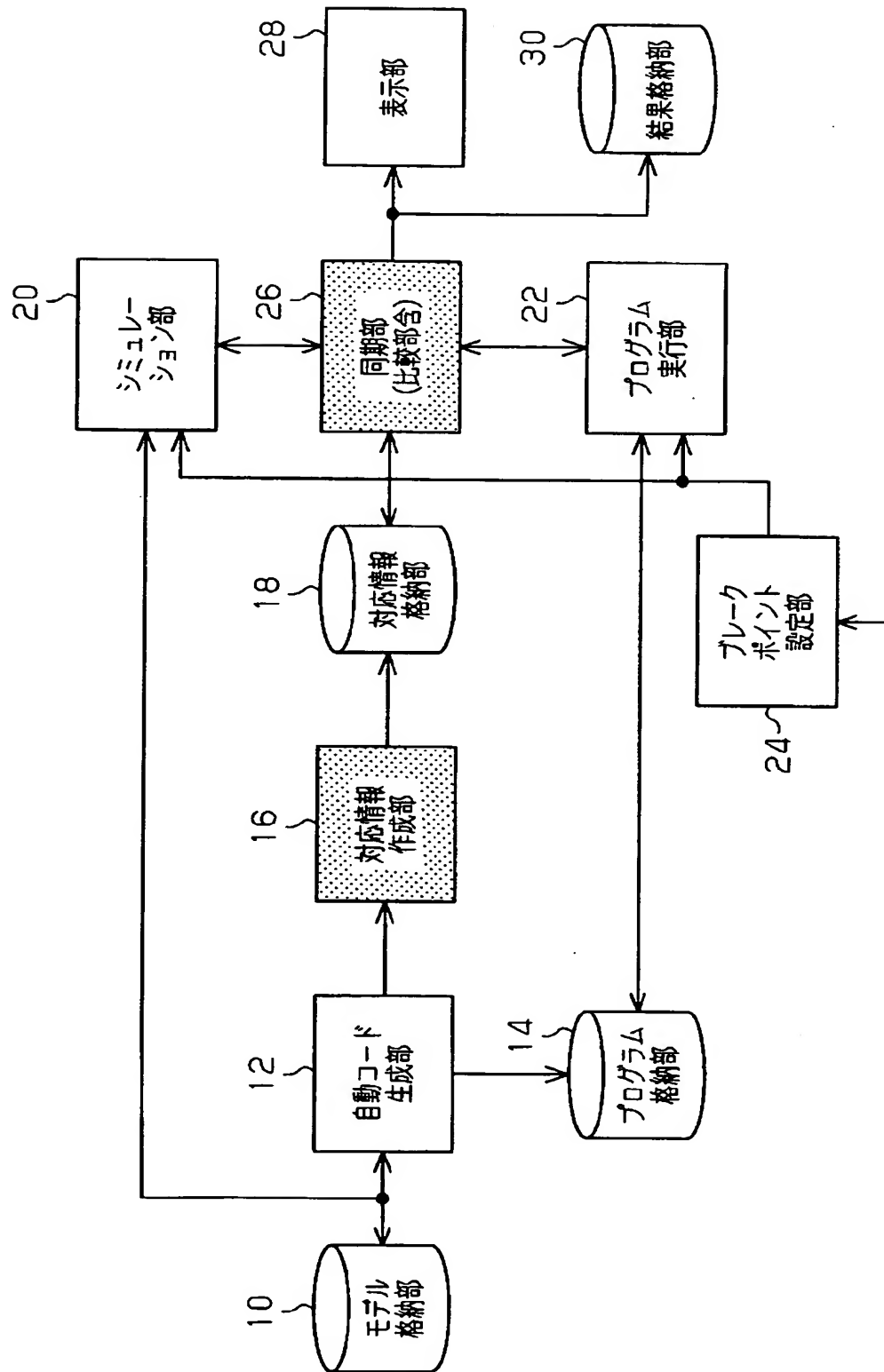
1 0…モデル格納部、1 2…自動コード生成部、1 4…プログラム格納部、1 6…対応情報作成部、1 8…対応情報格納部、2 0…シミュレーション部、2 2…プログラム実行部、2 4…ブレイクポイント設定部、2 6…同期部（比較部）、2 8…表示部、3 0…結果格納部。

【書類名】 図面

【図 1】

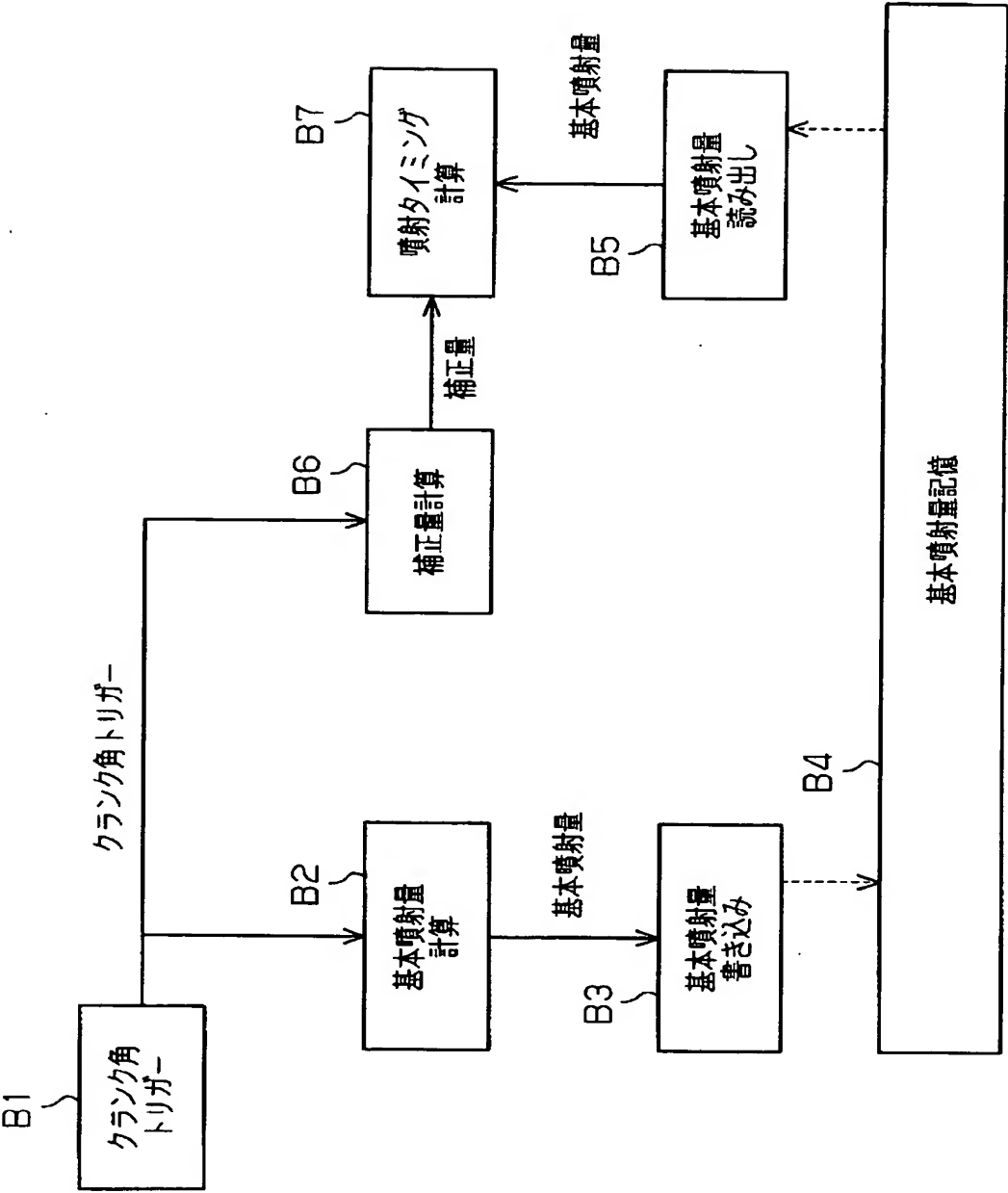


【図 2】





【図 3】



【図 4】

```

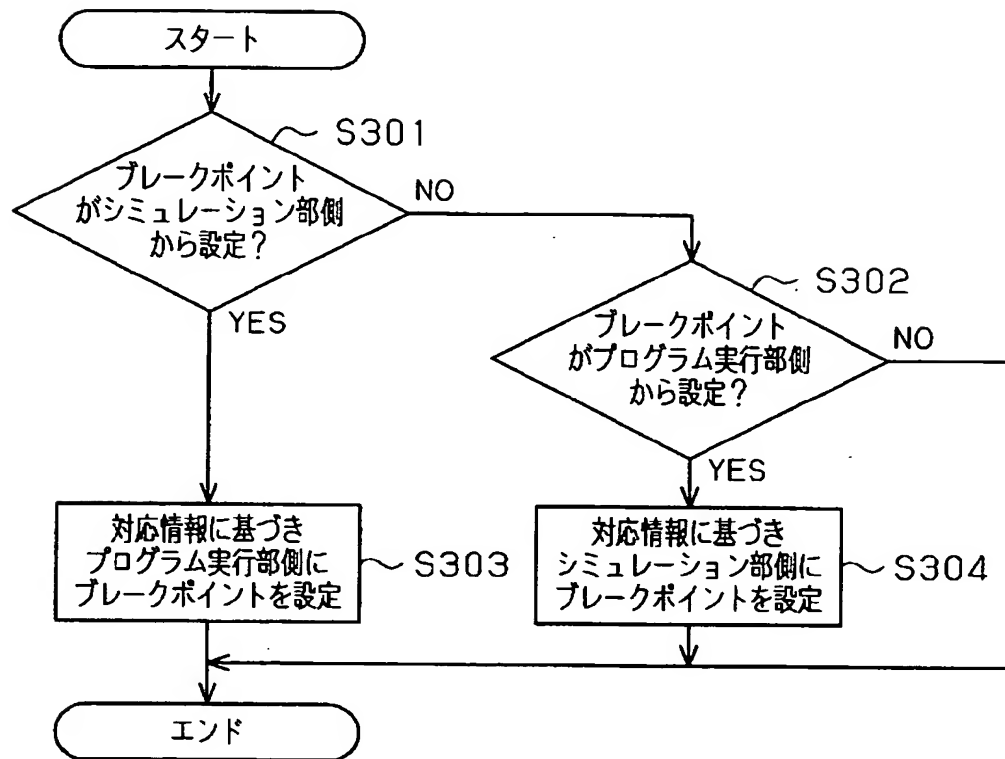
L1:
  //基本噴射量計算 (B2)
  v1<...;
L3:
  //補正量計算 (B6)
  v2<...;
L4:
  v3<read (m1); //基本噴射量読み出し (B5)
L5:
  //噴射量タイミング計算 (B7)
  v4<... (v2, v3);
L2:
  write (m1, v1); //基本噴射量書き込み (B3)

```

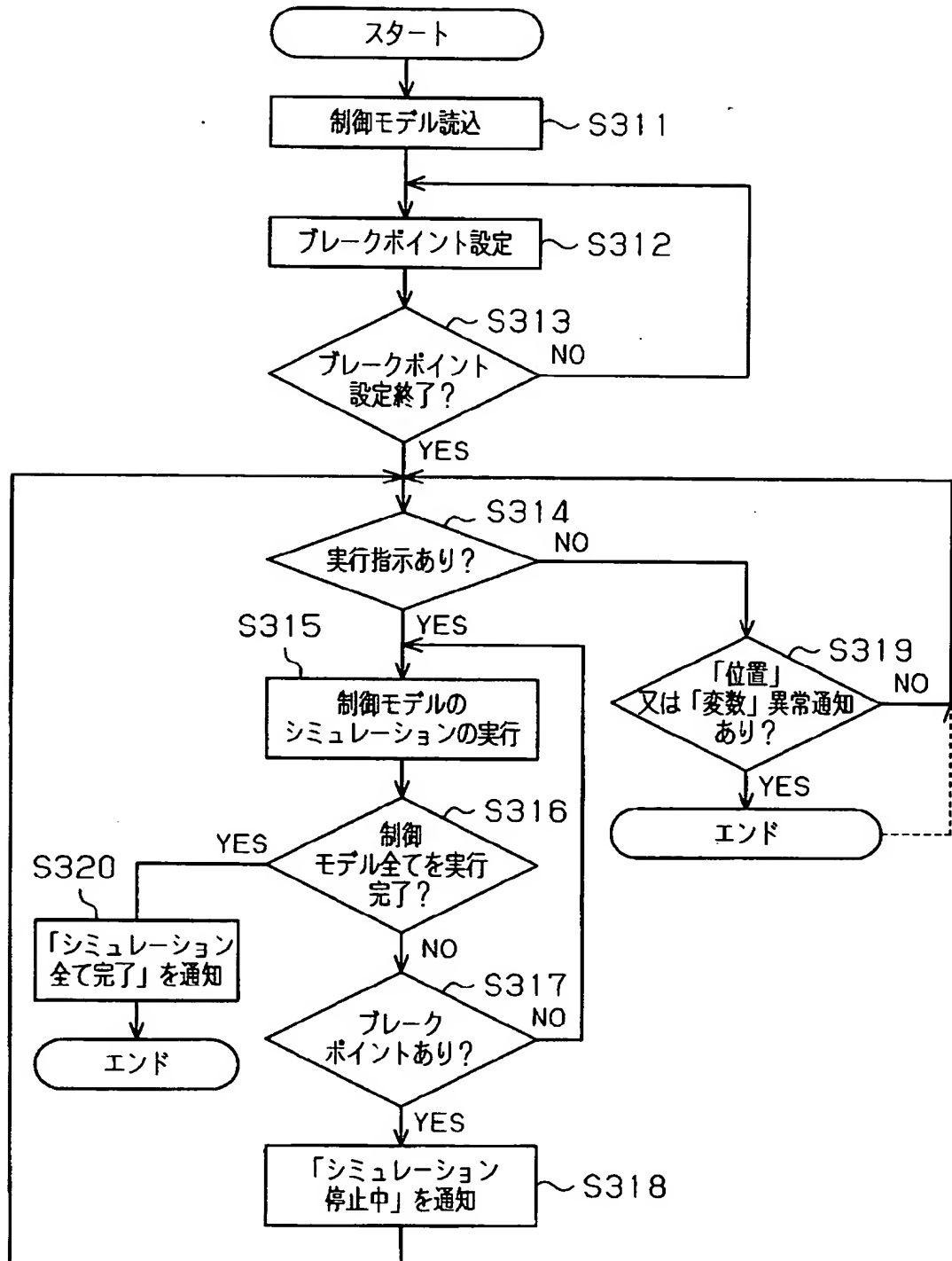
【図 5】

実行位置対応情報		変数対応情報	
ブロック	ラベル	ブロック結線/ストレージ	変数
クランク角トリガー (B1)			
基本噴射量計算 (B2)	L 1	基本噴射量	v 1
基本噴射量書き込み (B3)	L 2	基本噴射量	v 1
		基本噴射量記憶	m 1
補正量計算 (B6)	L 3	補正量	v 2
基本噴射量読み出し (B5)	L 4	基本噴射量	v 3
		基本噴射量記憶	m 1
噴射量タイミング計算 (B7)	L 5	補正量	v 2
		基本噴射量	v 3
		噴射量タイミング	v 4

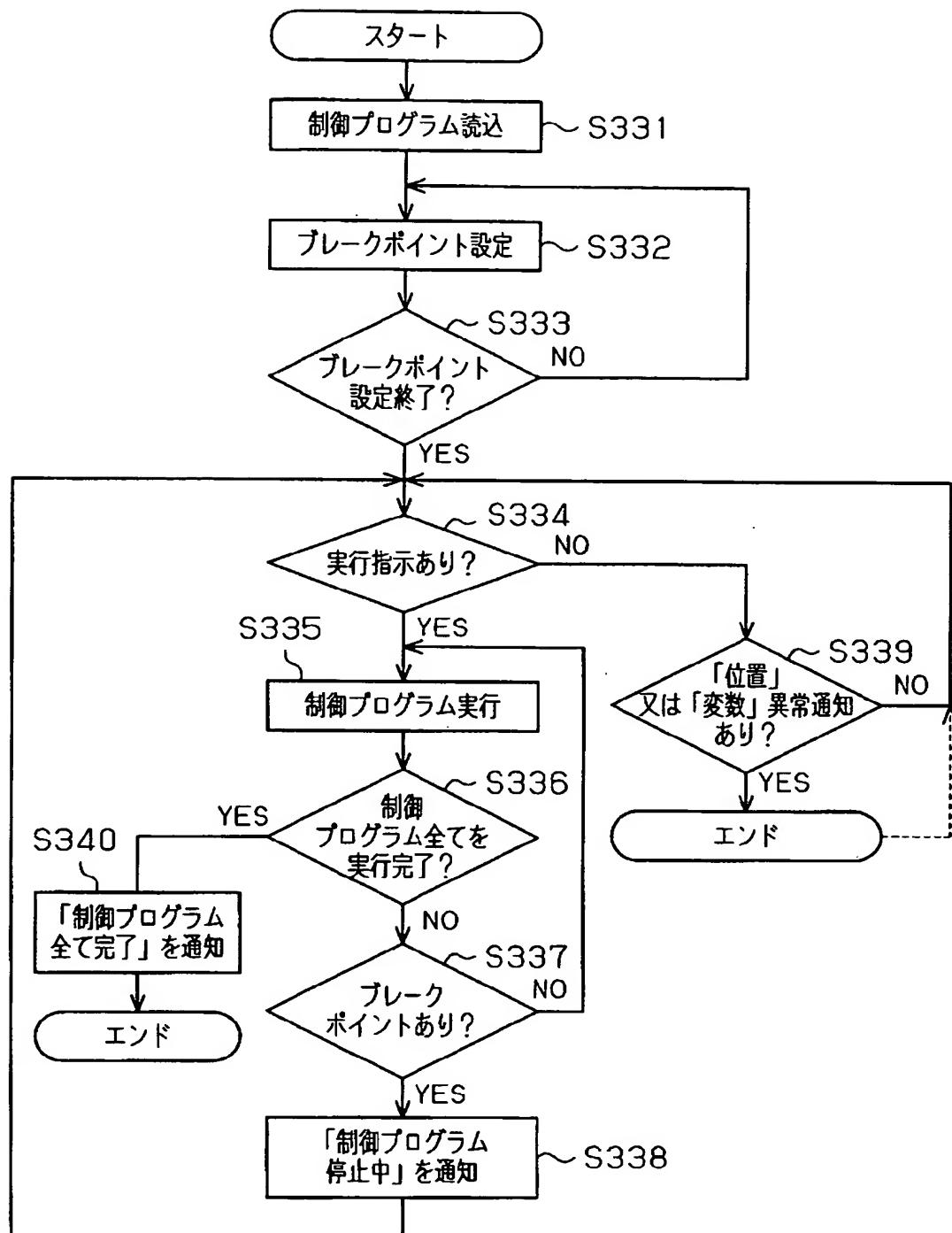
【図 6】



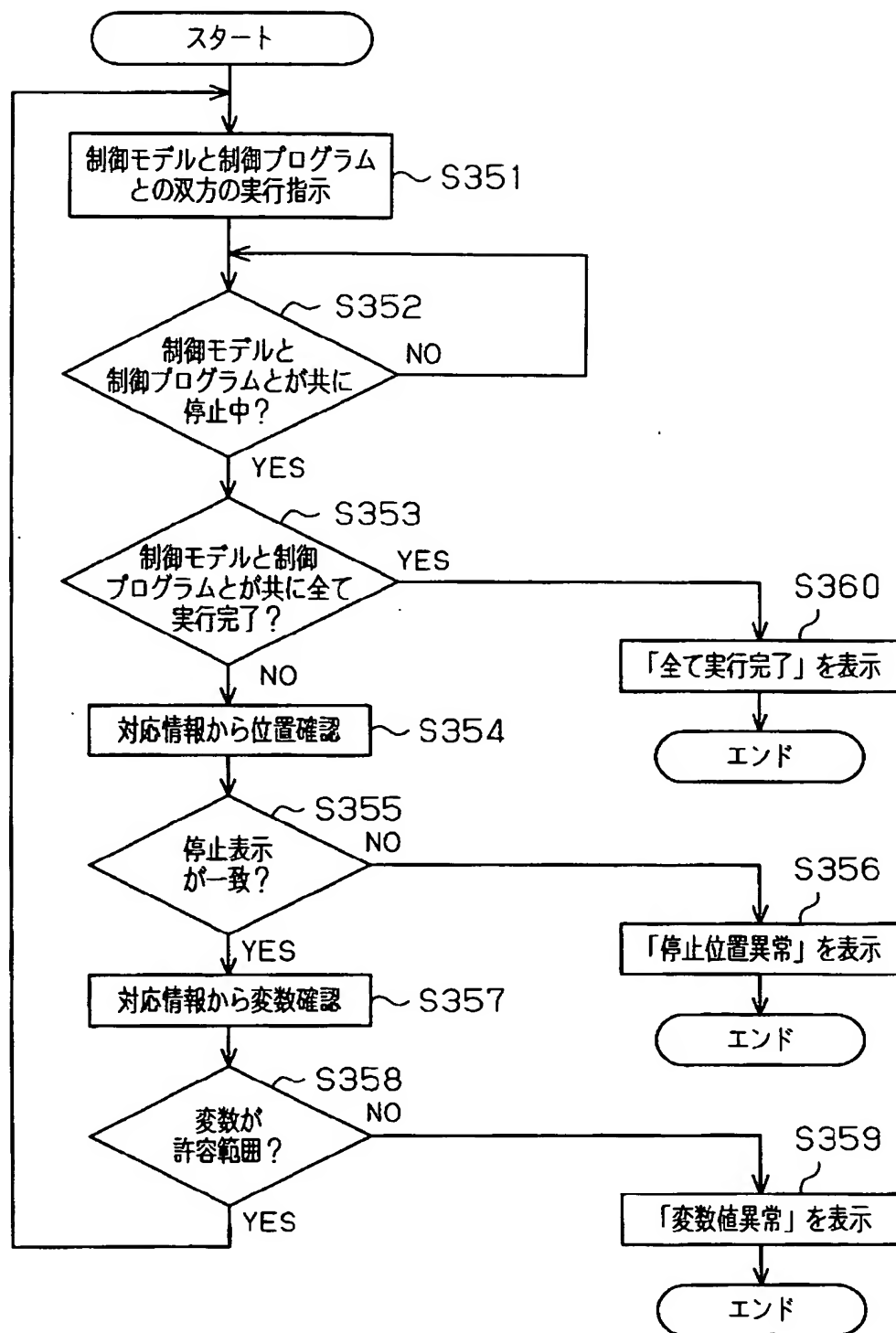
【図 7】



【図 8】



【図 9】



**【書類名】 要約書****【要約】**

**【課題】** モデルベース開発を行う場合にあって、制御プログラムや制御モデルの異常の有無の検査を簡易に行う。

**【解決手段】** 自動コード生成部 1 2 は、制御モデルを入力し、これから C 言語で記述された制御プログラムを生成して出力する。対応情報作成部 1 6 は、上記自動コード生成部 1 2 が制御プログラムを生成するに際して同自動コード生成部 1 2 の有する情報に基づき、制御モデルと制御プログラムとの対応関係を表す対応情報を作成する。シミュレーション部 2 0 は、制御モデルを入力し、その動作をシミュレーションする。また、プログラム実行部 2 2 は、上記制御プログラムを入力し、これを実行するものである。同期部 2 6 は、上記対応情報に基づきシミュレーション部 2 0 及びプログラム実行部 2 2 の動作結果を同期（関連付け）させる。

**【選択図】** 図 2

特願 2 0 0 3 - 4 2 3 5 8 3

出 願 人 履 歴 情 報

識別番号 [ 0 0 0 0 0 4 2 6 0 ]

1. 変更年月日	1 9 9 6 年 1 0 月 8 日
[変更理由]	名称変更
住 所	愛知県刈谷市昭和町 1 丁目 1 番地
氏 名	株式会社デンソー